LEVEL II

# Research Memorandum 77-2

# DESCRIPTION OF THE ASSIGNMENT ALGORITHM



Alison F. Fields

PERSONNEL ACCESSION AND UTILIZATION TECHNICAL AREA





U. S. Army

Research Institute for the Behavioral and Social Sciences

March 1977

Best Available Copy

Army Project Number
16 2Q762717A766

Manpower Utilization
Systems

9 Research Memorandum 77-2

6 DESCRIPTION OF THE ASSIGNMENT ALGORITHM

10 Alison F. Fields 14 ARI-RM-77-2

Bertha H. Cory, Work Unit Leader

12 40

Submitted by:
Ralph R. Canter, Chief
Personnel Accession and Utilization Technical Area

11 March 1977

Approved by:

E. Ralph Dusek, Director
Individual Training and Performance
Research Laboratory

J. E. Uhlaner, Technical Director
U.S. Army Research Institute for
the Behavioral and Social Sciences

| Accession For | |
|---|---|
| NTIS GRA&I | ☒ |
| DDC TAB | ☐ |
| Unannounced | ☐ |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or special |
| A | |

Research Memorandums are informal reports on technical research problems. Limited distribution is made, primarily to personnel engaged in research for the Army Research Institute.

408 010 JOB

## DESCRIPTION OF THE ASSIGNMENT ALGORITHM

Improved motivation and retention of a high quality officer corps within the Army will depend in part on the quality of the system which matches the Army's force structure requirements with the individual Army officer's career needs. In order to better understand the functioning of the current system and to conduct research on officer career progression, a computerized experimental research facility was designed and implemented at ARI. This facility (described by Van Nostrand in a report in preparation) allows ARI scientists to experiment (1) with systems for presenting career information and guidance tailored to the individual officer, (2) with the technology of personnel file review and updating, and (3) with other aspects of information system technology.

As part of this experimental system, a job assignment module was designed to allocate individuals to job categories according to each individual's suitability. The module is primarily the work of Dr. Robert Eastman (Eastman, 1977), based on the earlier work by Ford and Fulkerson (1957) and Granda and McMullen (1974). It is an assignment algorithm designed to reflect and be flexible to changes in Army policy and was developed with two potential users in mind:

1) Individual officers who would use it to make decisions in expressing their assignment preferences.

2) Officer Personnel Directorate management, particularly assignment officers, who would use it as an aid in making assignments.

Assignment polices are quantified and individuals' scores for jobs are determined in the following manner:

Step 1. The assignment officers determine the categories of jobs to which individuals will be assigned.

Step 2. Easily retrievable background variables which enter into the assignment decisionmaking process are identified.

Step 3. The assignment officers rate the overall importance of each of the background variables for each of the job categories, thus setting up a table of weights.

Step 4. For each background variable, a set of features exists which includes all the possible states describing an individual's background in that variable. Assignment officers weigh each of these features for its negative, positive, or null importance for each job category. These weights are consistent within a single variable; the relative importance among variables is reflected in the table of weights set up in Step 3.

Step 5. The appropriate product weights (feature weight x variable weight) summed across all background variables for each job category are determined for each individual officer from the available background data in order to obtain a utility score for each of the job categories. These utility scores are standardized to permit comparison among job categories.

Officers can be allocated to job categories in three ways: optimization of scores for assigned jobs for the total group; preselection of some individuals particularly suited for certain assignments and then optimization of the remainder; rank ordering of the individuals by their total scores (across job categories) followed by preselection and optimization.

## DATA AND TABLES OF WEIGHTS

Two kinds of information are necessary to operate the assignment algorithm: data on individuals, and tables of weights which determine how these data should be used in assignments. The first comes from the officers' records, while the second is determined by the assignment officers. This discussion will use examples from research application of the algorithm for Infantry and Quartermaster assignments.

### JOB CATEGORIES

The first step in setting up the tables of weights is to isolate the job categories. The assignment officer identifies basic types of jobs to which individuals must be assigned. For the Infantry, eight such categories were found: ROTC instructor (ROTC), civilian education (CE), long tour overseas (LT), short tour overseas (ST), CONUS[1] command (CC), CONUS staff (CS), Army school instructor (I), and reduction in force (RIF). The same categories were used for Quartermaster assignments with the exception of CONUS command; for Quartermasters the category of command (C) included both CONUS and overseas tours. One potential job category, military education, was not used in either the Infantry or the Quartermaster research. In both cases, the subject populations were the graduating classes of the advanced course and not due for another military education assignment.

### BACKGROUND VARIABLES

Next the assignment officer chooses the background variables which should influence the job category to which an individual officer should be assigned. The set of variables for the Infantry and Quartermaster officers were preferences for next assignment (PREF), civilian education level (CEL), component (COMP), manner of performance (MOP), time in service (TIS), and previous assignment history (EXP). Military education level was excluded because the military education level was the same for all individuals and thus had no predictive value.

---

[1] CONUS = Continental U.S.

## WEIGHTS OF BACKGROUND VARIABLES FOR JOB CATEGORIES

Each of these background variables has a different importance for each of the job categories. For example, for RIF, component was at one time very important because Regular Army officers were not vulnerable to RIF. However, recent changes in the law have removed this protection and now component is unimportant for RIF. To reflect the importance of the background variables in assigning individuals to jobs, each background variable is given a weight of importance for each job category in a table, such as that in Table 1. While a scale of 0-10 was used here, any positive scale ranging from zero to a three-digit number could be used. The program calls for six background variables and eight job categories, although "dummies" can be used if fewer of either are desired. A larger number of either would require changes to the program.

## FEATURES OF BACKGROUND VARIABLES AND THEIR WEIGHTS

The possible features of each of the background variables for determining job assignments must be defined. Each background variable can have as few potential features as desired; however, no more than 10 features can be used without changes to the program. A sample set of features appears in Table 2. These features need not be and usually are not numerical values; rather, many are simply descriptive statements of the individual's background. For the features, a second table of weights is determined by the assignment officer. The weights reflect relative importance of each feature within a variable for each job category. For example, the civilian education part of such a table is given in Table 3. These weights can take on negative and positive values, from a minus three digits to a plus four digits, although a scale of -10 to +10 was used here. For example, for an officer who has less than two years of college, the chances for an Army instructorship are very small; that particular Civilian Education Level counts against the job with a -5 weight. Each added educational degree increases the chances of getting an instructorship.

Without this system of double weighting, the only table needed would be a table of possible background variable features weighted for each job category. However, in this one table the importance of each possible feature would have to be decided upon relative to every other possible background variable feature. For example, having a master's degree would have to be weighted against a preference for CONUS staff jobs, having spent 80 months in the Army, etc., for each job. With the double weighting system, the weights of the features must be consistent only within the corresponding background variable and the assignment officer need not take the other background variables into account when weighting the features of one background variable. The weights of the background variables themselves must be relative to each other only within a job category.

Table 1

SAMPLE WEIGHTS OF BACKGROUND VARIABLES FOR JOB CATEGORIES

| Background Variables | Job Categories: ROTC Instructor (ROTC) | Civilian Education (CE) | Long Tour (LT) | Short Tour (ST) | CONUS Command (CC) | CONUS Staff (CS) | Instructor (I) | RIF |
|---|---|---|---|---|---|---|---|---|
| Preferences (PREF) | 5 | 5 | 5 | 3 | 7 | 8 | 7 | 1 |
| Civilian Education Level (CEL) | 9 | 5 | 3 | 3 | 2 | 4 | 6 | 3 |
| Component (COMP) | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 0 |
| Manner of Performance (MOP) | 10 | 10 | 5 | 5 | 8 | 6 | 8 | 9 |
| Time in Service (TIS) | 4 | 5 | 3 | 5 | 7 | 5 | 5 | 7 |
| Assignment History (EXP) | 10 | 7 | 5 | 7 | 6 | 7 | 8 | 5 |

Table 2

FEATURES OF THE SIX BACKGROUND VARIABLES USED IN THE QUARTERMASTER RESEARCH

Preference (PREF)

0 = None
1 = overseas
2 = command
3 = civilian education
4 = CONUS instructor
5 = CONUS staff
6 = preferences overriden by being on RIF list

Civilian Education Level (CEL)

1 = less than 2 years of college
2 = 2 or more years of college
3 = college graduate and/or a year or more of graduate school
4 = master's or professional degree
5 = Ph.D.

Component (COMP)

1 = Regular Army
2 = Other than Regular Army

Manner of Performance (MOP)

1 = upper third
2 = middle third
3 = lower third
4 = borderline cases between middle and lower third or extenuating circumstances for being in lower third

Time in Service (TIS)

1 = 95 or more months active federal commissioned service
2 = 80-94 months
3 = 65-79 months
4 = 64 or fewer months
5 = over 156 months active federal service

Assignment History (EXP)

1 = no command; not up for a short tour
2 = two or more short tours or just back from a short tour
3 = zero overseas or 72 months since a single short tour
4 = just back from a long tour
5 = RIF list or one time passover for promotion; overrides other factors
6 = needs a long tour

Table 3

SAMPLE WEIGHTS OF CIVILIAN EDUCATION
LEVEL FEATURES FOR JOB CATEGORIES

| Civilian Education Level (CEL) | Job Categories | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ROTC | CE | LT | ST | CC | CS | I | RIF |
| less than 2 years of college | -10 | 0 | 0 | 0 | 0 | 0 | -5 | 3 |
| 2 or more years of college | -10 | 10 | 0 | 0 | 0 | 0 | 0 | 1 |
| college graduate and/or a year or more of graduate school | 5 | 3 | 0 | 0 | 0 | 0 | 6 | 0 |
| master's or professional degree | 6 | -9 | 0 | 0 | 0 | 0 | 8 | -1 |
| Ph.D. | 10 | -10 | 0 | 0 | 0 | 0 | 10 | -2 |

INDIVIDUAL DATA

It may be noticed in Table 2 that each feature of a background variable is assigned a different number from the weights in Table 3. This number is simply a convention for identification, much as football players are assigned numbers. In determining an individual officer's data, the personnel records are carefully studied and the identifying numbers which best describe each situation are assigned. Thus each officer's data consist of some sort of identifier and six numbers that identify the feature of each background variable which is descriptive of this individual.

SUMMARY

Three sets of information have been identified:

1. The individuals' data--the codes representing the particular features of the background variables (see Table 2).

2. The table of weights of background variables for job categories--the measures of the overall importance of the different general kinds of background experience in determining assignments (see Table 1).

3. The table of weights of features of background variables for job categories--the relative importance of the different features of a background variable for each job category (see Table 3).

## OTHER PARAMETERS

### JOB QUOTAS

Ten other parameters must be specified. One parameter is the set of job quotas, the number of jobs available within each job category. The algorithm requires that the total number of jobs be equal to the number of individuals assigned.

### NUMBER OF FEATURES, NUMBER OF INDIVIDUALS, NUMBER OF JOB CATEGORIES

As discussed earlier, each background variable can have up to 10 features; the set of the actual number of features of each of the six background variables must be entered as the second parameter. The total number of individuals (up to 200 without changes in the program) assigned is the third parameter, the number of job categories the fourth.

### DESCRIPTION OF PRESELECTION: IDIF, KDIF, IADDIF

The next parameters require discussion on how the algorithm works. Three allocation methods can be run. The first optimizes scores across job categories. The second and third preselect certain individuals and then assign the rest. The preselection takes place as follows:

First, scores are calculated for each individual for each job. For each category, this is done by multiplying the overall weight of each background variable by the weight associated with the individual's feature for the corresponding background variable. The products are then summed to produce a single score for the job category. This process is repeated using the appropriate weights for each of the job categories. After determining the scores of all officers assigned for each of the job categories, the scores are standardized as follows: The range of possible (not actual) scores received by the officers within each job category is determined. The minimum possible score for that job category is also determined. Then the minimum score is subtracted from the officer's score for a job category, the answer is multiplied by 1000, and that product is divided by the range. Preselection uses these standardized scores.

The basic idea of preselection is to allocate officers who are especially well qualified for a certain job to that job. In optimizing scores, there is no guarantee that individuals will be assigned to their highest scored jobs. An officer is "especially well qualified" for job A, as defined by the assignment algorithm, if the score for job A is the highest score that that individual has <u>and</u> it is greater than any other score of that same individual by at least an amount called IDIF. IDIF is a parameter defined by the assignment officer. It can be any positive number. IDIF is set at whatever value the user feels is appropriate to differentiate officers particularly well suited for one job category from officers who might be reasonably equally suited for several job categories.

The algorithm allows the user to increase IDIF automatically during the running of the program, putting out a set of assignments at each value of IDIF. This is done by setting the parameter IADDIF to the amount by which the user wants IDIF to be increased for each allocation cycle. A third parameter KDIF is then set as the maximum value the experimenter wishes IDIF to reach before ending the program. If only one allocation is desired IADDIF can be set to 0 and KDIF can be set equal to IDIF. IDIF may be thought of as a threshold for preselection, IADDIF as a just noticeable difference (jnd), and KDIF as a maximum.

## QUOTAS FOR OPTIMIZATION

In running the assignment algorithm, the user also has the option of limiting the number of officers who can be assigned by preselection for each group. This is done by entering, as an eighth parameter, minimum quotas for the job categories which must be assigned by the optimization method. Then the preselection subroutine will assign qualified individuals until the quotas are met; any qualified individual found after quotas are met is ignored by the preselection subroutine. If not enough individuals have scores greater by IDIF for a job category, the algorithm makes no attempt to meet the quotas; the quotas are a limit, not a goal.

## IOPTIO

One way to run the algorithm without preselection is to make the minimum quotas assigned by optimization equal to the quotas for the job categories themselves. However, this would not bypass the preselection subroutine; it would just make it futile. The more efficient method is to use the IOPTIO parameter: if IOPTIO=1, allocation is made using both preselection and optimization subroutines; if IOPTIO=2, allocation is done using only the optimization subroutine; if IOPTIO=3, individuals assigned are rank-ordered first and then allocated by preselection and optimization.

The third IOPTIO option should be discussed further. The rank-ordering is performed by adding together an individual's scores across the first seven job categories and then rank-ordering the summed scores. Objective is to rank-order people according to their overall merit in the belief that a person more fitted to many jobs is somehow "better" than one less fitted. Because RIF is a negative job--that is, people well-fitted for RIF are less well-fitted for other jobs--the algorithm does not add the scores for the eight job category, RIF, when performing this rank ordering.

One other peculiarity of this third option should be mentioned: when scores are rank-ordered and then allocated by the preselection subroutine, the "best" and "worst" individuals are tested for preselection first. In other words, the preselection subroutine does not start at the beginning of the rank-ordered group and go to the end. Instead, it proceeds in the following manner.

The subroutine takes a matrix of individual ID numbers and scores, such as:

| Individual ID No. | Individual Scores by Job Category | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ROTC | CE | LT | ST | C | CS | I | RIF |
| 1 | 625 | 937 | 808 | 529 | 211 | 721 | 324 | 034 |
| 2 | 802 | 209 | 601 | 595 | 492 | 600 | 734 | 021 |
| 3 | 035 | 534 | 603 | 649 | 211 | 529 | 804 | 321 |
| 4 | 023 | 769 | 502 | 324 | 211 | 742 | 069 | 926 |
| 5 | 426 | 321 | 021 | 226 | 301 | 230 | 123 | 993 |

This subroutine then checks the first individual. If a score exists at least IDIF larger than any other score for this individual, and if slots are still available in that job category for filling by preselection, the following occurs: The subroutine assigns the individual to that job category; decrements number of slots available in that job category; decrements the number of people to be assigned; and exchanges the row of this individual in the matrix with the last row in the matrix not yet assigned. The subroutine does not move to the next row of the matrix until the new top row has been rejected for preselection. For example, in the matrix above during the first pass, the subroutine would look at the first row and see that the CE score was at least 129 points greater than all other scores. Let us say the IDIF equals 100 and the CE quota is still open. Then the matrix after the first pass would look like this:

| Individual ID No. | Individual Scores by Job Category | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ROTC | CE | LT | ST | C | CS | I | RIF |
| 5 | 426 | 321 | 021 | 226 | 301 | 230 | 123 | 993 |
| 2 | 802 | 209 | 601 | 595 | 492 | 600 | 734 | 021 |
| 3 | 035 | 534 | 603 | 649 | 211 | 529 | 804 | 321 |
| 4 | 023 | 769 | 502 | 324 | 211 | 742 | 069 | 926 |
| 1 | 625 | 937* | 808 | 529 | 211 | 721 | 324 | 034 |

*Assignment has been made

Now we see that for officer 5, the RIF score meets the criteria; if the RIF quota has not yet been filled, the matrix changes to:

| Individual ID No. | Individual Scores by Job Category | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ROTC | CE | LT | ST | C | CS | I | RIF |
| 4 | 023 | 769 | 502 | 324 | 211 | 742 | 069 | 926 |
| 2 | 802 | 209 | 601 | 595 | 492 | 600 | 734 | 021 |
| 3 | 035 | 534 | 603 | 649 | 211 | 529 | 804 | 321 |
| 5 | 426 | 321 | 021 | 226 | 301 | 230 | 123 | 993* |
| 1 | 625 | 937* | 808 | 529 | 211 | 721 | 324 | 034 |

* Assignment has been made

Now the RIF score again meets the criteria, but let us say the quota for RIF preselection is complete. Therefore, the matrix remains as is and we now look at the next row, officer 2. No score meets the criterion, so the matrix remains the same. The I score for officer 3 meets the criteria. However, as there is no other row which has not been assigned, the matrix remains the same although the number of people to be assigned by optimization is decremented. The optimization subroutine behaves as if the matrix of scores had only as many rows in it as the number of people remaining to be assigned--in this case, two.

## JPRINT

The final parameter which must be entered into the program is the JPRINT option. If the JPRINT option equals 1, all entered data, the raw (non-standard) scores, and the standardized scores and program results are printed out. If JPRINT=2, only the standardized scores and program results are printed.

## SUMMARY

Each of these different parameters must be entered into the program before it is run:

Job Quotas
The number of features of each background variable
Number of individuals
Number of job categories
IDIF
IADDIF
KDIF

Minimum number of slots within each job category to be assigned by optimization
IOPTIO
JPRINT

The format for the data and the parameters can be found in Appendix A, Program Reference Sheets.

## DESCRIPTION OF THE ASSIGNMENT ALGORITHM PROGRAM

The assignment algorithm is a FORTRAN program which runs on the UNIVAC 1108 with the EXEC 8 operating system (See Appendix B, Program Listings). The main program, MODOTT, controls the calling of the five subroutines: INPUT, OTT, PRESEL, RANKR, and CALCU.

The first subroutine, called INPUT, reads the data and the parameters from cards and calculates the raw and standardized scores. If JPRINT=1, INPUT prints out the raw data and raw scores; if JPRINT=2, this printing is skipped. In either case the table of standardized scores is printed out. INPUT also calculates and prints out the means and standard deviation of the scores for each job category.

After INPUT, MODOTT checks IOPTIO and calls RANKR if IOPTIO=3. RANKR rank orders the individual data rows according to the total scores (not including the eighth column, RIF scores), as discussed earlier. If IOPTIO≠3 or after RANKR, MODOTT assigns ID numbers to each individual and prints out the job quotas and the IOPTIO option. If IOPTIO=1 or =3, PRESEL is called; otherwise MODOTT skips to OTT.

The manner in which PRESEL allocates officers has already been described in the discussion of the third IOPTIO option. The printed output of the PRESEL subroutine is a set of two lists: one consists of the identifiers and the two highest scored job categories of those officers who were preselected for certain assignments and the other consists of the identifiers and two highest scored job categories for those to be assigned by optimization. After assigning as many officers as consistent with the scores and the quotas, PRESEL returns to MODOTT which immediately calls OTT to finish the allocation by optimization.

OTT is the optimization subroutine. It is based on the Ford-Fulkerson algorithm (1957) which has been modified over the years by several people in the ARI computer center (Granda and McMullen, 1974). It works on a classification problem where there are N individuals to be assigned to J job categories where N is greater than J, each job category has a quota $Q_j$ of individuals to be assigned to it, the sum of all the quotas is equal to N ($\sum_{j=1}^{j} Q_j = N$), and no individual can be assigned to more than one job category.

Let $i$ refer to the individual number
$j$ refer to job category number
$S_{ij}$ refer to the score of individual $i$ for job $j$
$X_{ij}$ refer to an index which is set to 1 when individual $i$ is assigned to job $j$ and to 0 when individual is not assigned to job $j$.

OTT then maximizes M:

$$M = \sum_{j=1}^{j} \sum_{i=1}^{N} X_{ij} \; S_{ij}$$

CALCU is called to find the allocation average for the set of assignments by adding the individuals' scores for jobs they were actually assigned to and dividing by the number of individuals. The allocation average is a measure of how well the optimization has worked.

MODOTT then prints out the final assignments and the allocation average. It adds IADDIF to IDIF, and if IDIF is less than or equal to KDIF, MODOTT goes back to the point at which ID numbers are assigned and begins again. If IDIF is greater than KDIF, the program ends.

## DISCUSSION

Although the assignment algorithm was designed and used for modeling the behavior of an assignment officer, it is not limited to this application. The algorithm can be applied to any case which fits the double weighting scheme. Indeed, the second set of weights (those of the features of background variables for job categories in these applications) would not have to be "weights" in the sense used so far. The weights could be, for instance, scores or percentiles, although this would be most practical if scores or percentiles were limited to ten or less.

The assignment algorithm does not assign individuals against specific requisitions. Postal officer, mortuary officer, and personnel officer all might fall under "CONUS staff." Assignments to Fort Bragg, Fort Benning, and Fort Bliss might all be "command." This non-assignment to the actual job is not as great a failing as it might seem. After officers reach their new post, they are often assigned other jobs based on the commander's view of the unit's needs. However, it is theoretically possible to increase job categories to deal with actual jobs. This would simply require very large matrices of weights, very large computers, and very long run times. See Granda and McMullen, 1974, for a discussion of this problem.

Assigning officers to specific geographic locations (e.g., Fort Hood, Fort Jackson) might be accomplished by addition of a subroutine to maximize number of officer geographic preferences matched with specific locations, after the officers had been assigned to appropriate job categories.

Even without ability to assign against specific requisitions, the assignment algorithm could be used by assignment officers as a first pass to divide officers into tentative groups of assignment categories from which the assignment officer could work. This might be particularly helpful where large numbers of assignments are being processed at once, such as in large branches or first assignments of second lieutenants.

A possible use of the assignment algorithm by individual officers within the career progression system framework is shown in Figure 1. In this case, the algorithm would be one module within a career progression system running on a computer accessible by both management (assignment officers) and individual officers, although it is not necessary for either to have access to all of the system. Under this scheme, individuals, having discovered their probable job categories (which might be something like the two job categories which the subroutine PRESEL produces) could have feedback into the assignment process by reevaluating their preferences before these preferences are fed into the final running of the algorithm.

If the assignment algorithm were not adopted for use by management in the actual assigning of jobs, it might still be useful within the career progression system as a tool for individuals to use to predict possible assignments they might receive. Perhaps this could be done prior to filling out preference forms for next assignment.

The algorithm would be more easily used if the individuals' data did not have to be coded by hand. If the computer could read an individual's file and code it properly for the algorithm, the most time-consuming part of the process would disappear. Since much of the personnel data in the Army is computerized, this is not impossible. For example, the computer could code the time in service variable from the Officer Master Tape Record. Assignment history would be much more difficult to code, but if the algorithm is to be used, the effort would be worth it.

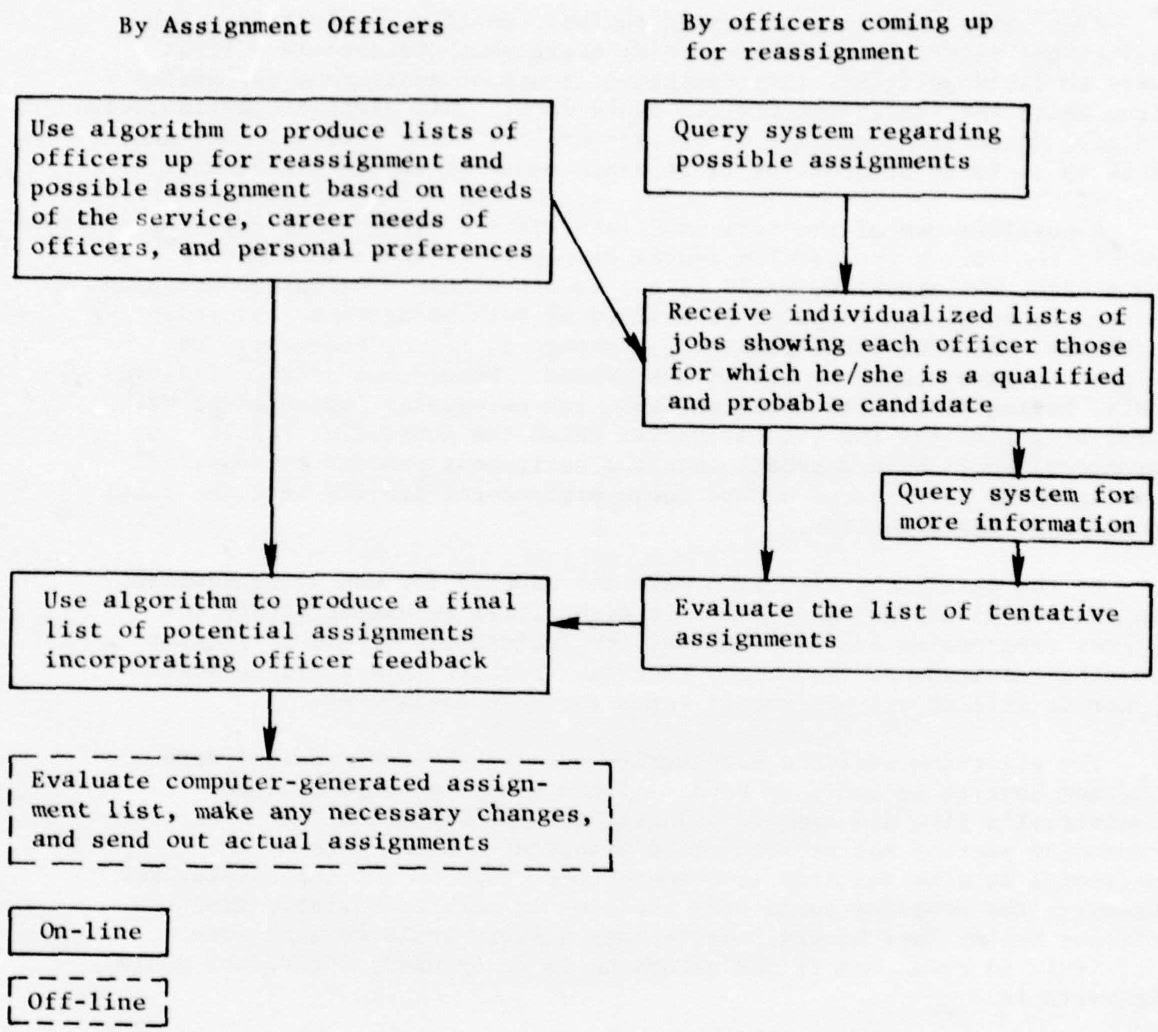



Figure 1. Possible use of the algorithm

## REFERENCES


Eastman, R. F. Development of a computer-assisted assignment system for Army officers--The assignment module. ARI Technical Paper (in press).

Fields, A. F. Application of the assignment algorithm to Quartermaster captains. ARI Research Memorandum 77-3. March 1977

Ford, L. R., and Fulkerson, D. R. Solving the transportation problem. The RAND Corporation, 1957.

Granda, T. M., and McMullen, R. L. A heuristic approach to a large personnel assignment problem. Paper presented at the Fall ORSA/TIMS National Convention, 17 October 1974.

Van Nostrand, S. IDEAS--Interactive Data-driven Experimental Army System. ARI Research Memorandum, in preparation.

APPENDIXES



PRECEDING PAGE BLANK

APPENDIX A

## PROGRAM REFERENCE SHEET

PROGRAM NUMBER Z600 MODIFICATION NO. A

PROGRAMMER Robert Eastman

PROGRAM TITLE ASGALG

MODIFIER Alison Fields

DATE 18 September 1975

PURPOSE ASGALG is an alocation algorithm for use in experimental modeling of Army assignment policies. It allows the manipulation of background variables, weights, and job categories. Allocation can be made in three ways: optimizing the utility scores for jobs across the whole group; preselecting for assignment individuals particularly suited for one job rather than others and then optimizing scores for the rest of the group; rank ordering the individuals before preselection.

MEMORY PROTECT

TIME SHARE

SUBROUTINES CALLED BY PROGRAM

INPUT
RANKR
PRESEL, OTT CALCU

MATH FORMULAE USED

SET-UP (DIAGRAM) (See attached sheet)

FORMAT OF INPUT DATA

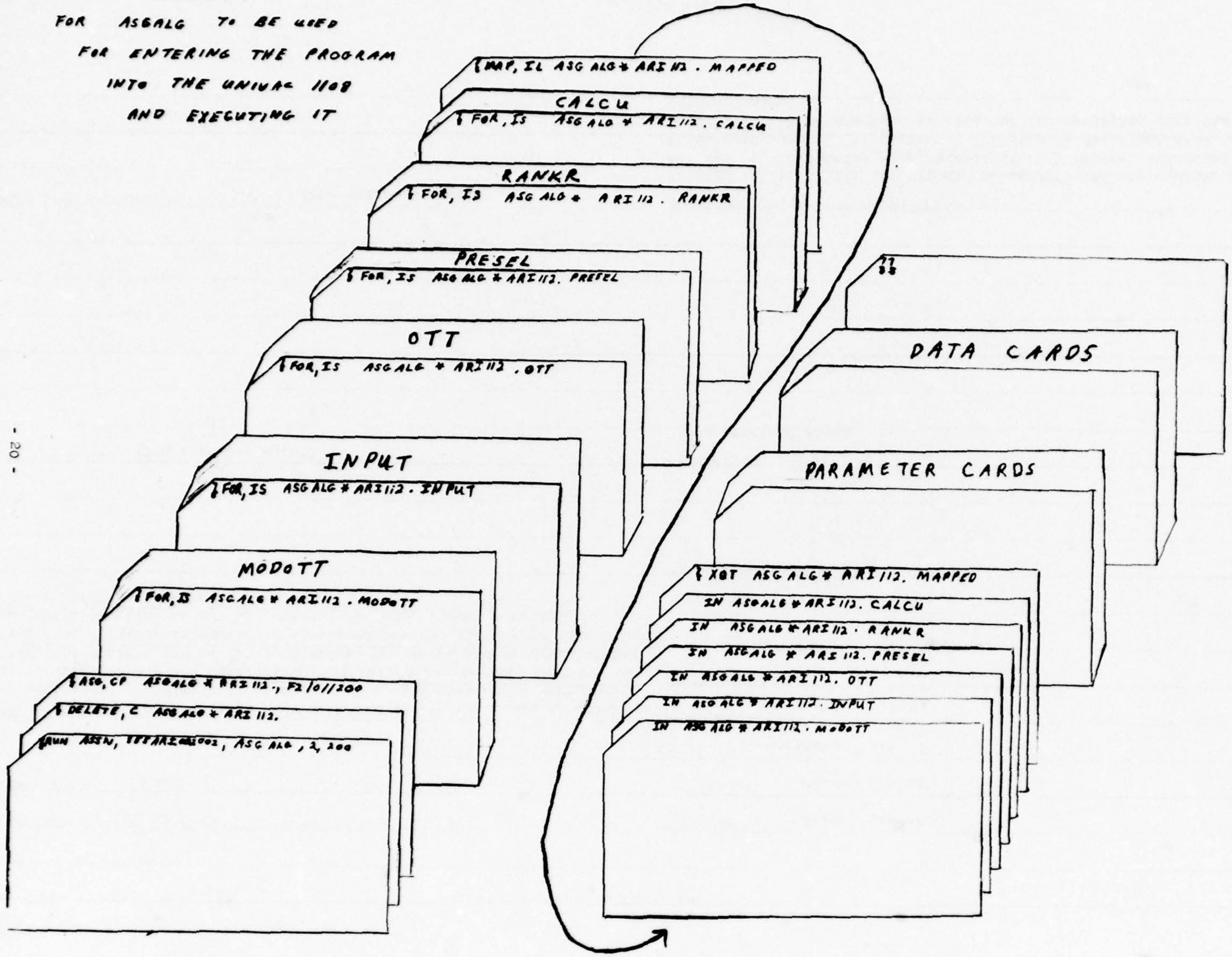
(See attached sheet)

...ROL CARDS (PARAMETERS, ETC) (See attached sheet)

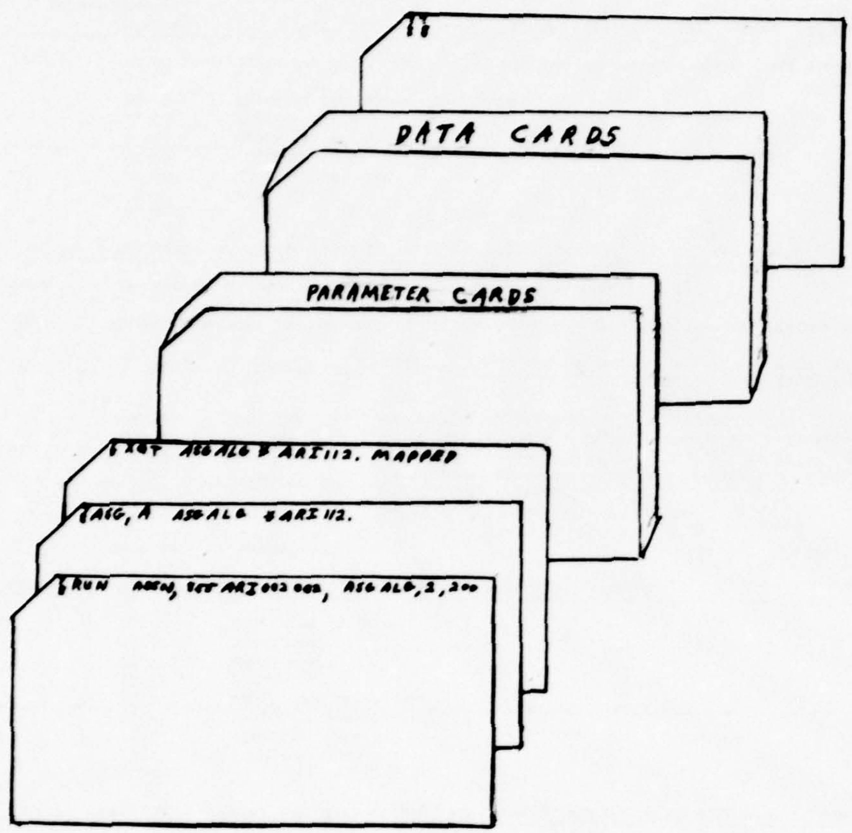
TYPE OF OUTPUT (DESCRIPTION)

Printed tables of identifiers and normalized utility scores, ID number and two job categories with highest utility scores, and identifiers and final assignments. If parameter JPRINT = 1, the input data and some working variables are also printed out for verifying purposes. See attached sample.

SET-UP DIAGRAM
FOR ASGALG TO BE USED
FOR ENTERING THE PROGRAM
INTO THE UNIVAC 1108
AND EXECUTING IT



SET- UP DIAGRAM
FOR ASGALG TO BE WED
WHEN THE PROGRAM
IS ALREADY IN THE
UNIVAC 1108



Control Cards (Parameters, Etc.)

Card #1 Minimum number of job slots within each category which must be allocated by subroutine OTT. All fields should be right justified.

| Columns | Job Category |
|---|---|
| 1-5 | ROTC instructor |
| 6-10 | Civilian education |
| 11-15 | Long tour (overseas) |
| 16-20 | Short tour (overseas) |
| 21-25 | Command |
| 26-30 | CONUS staff |
| 31-35 | Instructor |
| 36-40 | Reduction in Force |

Card #2 Main parameter card. All fields should be right justified.

| Columns | Mnemonics | Definition |
|---|---|---|
| 1-5 | MEN | number of individuals to be assigned |
| 6-10 | LC | number of job categories |
| 11-15 | IDIF | the minimum amount of difference between two highest scores for qualification for PRESEL allocation |
| 16-20 | KDIF | the maximum value IDIF can take on; set to IDIF if only one allocation cycle is desired |
| 21-25 | IADDIF | the amount by which IDIF is increased for each allocation cycle |
| 26-30 | IOPTIO | options for allocation: 1 = PRESEL and OTT<br>2 = OTT only<br>3 = rank order first by scores then PRESEL and OTT |
| 31-35 | JPRINT | option to print out raw data 1 = PRINT 2 = NO PRINT |

Card #3 Column constants for use in OTT. If not known 0's should be entered. Format same as card 1.

Card #4 Job quotas for each job category. Format same as card 1.

Card #5 The number of offsets used by each background variable. All fields should be right justified.

| Columns | Background Variable |
|---|---|
| 1-5 | Preferences |
| 6-10 | Civilian Education |
| 11-15 | Component |
| 16-20 | Manner of Performance |
| 21-25 | Time in Service |
| 26-30 | Previous Assignment Experience |

Card #6 Preference weights for ROTC. Zero (or a blank) is the default value. All fields should be right justified.

| Columns | 1-5 | 6-10 | 11-15 | 16-20 | 21-25 | 26-30 | 31-35 | 36-40 | 41-45 | 46-50 |
|---|---|---|---|---|---|---|---|---|---|---|
| Offset number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |

Card #7 Preference weights for CE, format same as card 6.
Card #8 Preference weights for LT, format same as card 6.
Card #9 Preference weights for ST, format same as card 6.
Card #10 Preference weights for C, format same as card 6.
Card #11 Preference weights for CS, format same as card 6.
Card #12 Preference weights for I, format same as card 6.
Card #13 Preference weights for RIF, format same as card 6.

Cards # 14-21 Civilian Education weights for the eight job categories. Format same as cards 6-13.
Cards # 22-29 Component weights for the eight job categories. Format same as cards 6-13.
Cards # 30-37 Manner of Performance for the eight job categories. Format same as cards 6-13.
Cards # 38-45 Time in Service for the eight job categories. Format same as cards 6-13.

Cards # 46-53 Previous Assignment History for the eight job categories. Format same as cards 6-13.

Cards # 54-57 Overall importance (weights) of the background variables for the job categories.

| | | | | |
|---|---|---|---|---|
| Card 54 | columns 1-30 | ROTC weights | columns 31-60 | CE weights |
| Card 55 | columns 1-30 | **LT** weights | columns 31-60 | ST weights |
| Card 56 | columns 1-30 | C weights | columns 31-60 | CS weights |
| Card 57 | columns 1-30 | I weights | columns 31-60 | RIF weights |

Format (all fields right justified)

| | | | | | | |
|---|---|---|---|---|---|---|
| Columns | 1-5 | 6-10 | 11-15 | 16-20 | 21-25 | 36-30 |
| | 31-35 | 36-40 | 41-45 | 46-50 | 51-55 | 56-60 |
| Job categories | PREF | CEL | COMP | MOP | TIS | EXP |

(See attached sample parameters)

```
     0     2     4     1     4     3     0     1
    57     8   100   100   100     3     1
     0     0     0     0     0     0     0     0     0     0     0
     0     8    13     5    13    12     0     6
     7     5     2     4     5     6
     0     0     2     5     0   -10
     0     0     6     0     0   -10
     8     0     0     0     0    -8
    10     0     0     0     0    -8
     0    10     0     0     0   -10
     0     5     0     0     7    -5
     0     0     3     8     0    -5
     0     0     0     0     0    10
   -10   -10     5     6    10
     0    10     3    -9   -10




    -5     0     6     8    10
     3     1     0    -1    -2

     5






    10     5
    10     5


     5

    10     5
     0     0    10     5

     5     2     0     0    -9


    10     7     5     2

     6     4     3     2
     5     5     5     5
    -5     0     0     5   -10     4
    -5     0     5     0   -10     2
     5    -3     8    -5   -10    10
     0   -10    10    -5   -10     0
     9     9     0     8   -10     2
     6     9    -5    10     5     7
     3     6     0     5     5     5
     0     0     0     0    10
     5     9     3    10     4    10     5     5     3    10     5     7
     5     3     2     5     3     5     3     3     2     5     5     7
     7     2     2     8     7     6     8     4     2     6     5     7
     7     6     2     8     5     8     1     3     0     9     7     5
••
```

Sample Parameters

(Missing lines are all zeros.)

Format of Input Data

Cards #1 to 1 & N — Individual Data Cards

| Columns | Definition |
|---|---|
| 1-15 | identifier, label or name |
| 16 | offset for PREF |
| 17 | offset for CEL |
| 18 | offset for COMP |
| 19 | offset for MOS |
| 20 | offset for TIS |
| 21 | offset for EXP |

(See attached sample data)

```
OFFICER A      132236
OFFICER B      242241
OFFICER C      231241
OFFICER D      522154
OFFICER E      131226
OFFICER F      131233
OFFICER G      231211
OFFICER H      631335
OFFICER I      232213
OFFICER J      131136
OFFICER K      332136
OFFICER L      112154
OFFICER M      522143
OFFICER N      632335
OFFICER O      131126
OFFICER P      132426
OFFICER Q      331134
OFFICER R      132136
OFFICER S      231141
OFFICER T      132436
OFFICER U      131136
OFFICER V      232131
OFFICER W      131236
OFFICER X      532132
OFFICER Y      232336
OFFICER Z      132126
OFFICER AA     112452
OFFICER BB     931132
OFFICER CC     542232
OFFICER DD     241241
OFFICER EE     231234
OFFICER FF     132313
OFFICER GG     132336
OFFICER HH     232331
OFFICER II     332126
OFFICER JJ     531332
OFFICER KK     242131
OFFICER LL     132113
OFFICER MM     131126
OFFICER NN     222343
OFFICER OO     532232
OFFICER PP     631325
OFFICER QQ     131423
OFFICER RR     232231
OFFICER SS     331136
OFFICER TT     532122
OFFICER UU     232131
OFFICER VV     632325
OFFICER WW     612325
OFFICER XX     132116
OFFICER YY     222136
OFFICER ZZ     231311
OFFICER AAA    542111
OFFICER BBB    931126
OFFICER CCC    532332
OFFICER DDD    631335
OFFICER EEE    931212
••
```

Sample Data

APPENDIX B

```
C     PROGRAM MODOTT
C  MODOTT IS THE MAIN PROGRAM
C       KO = MATRIX OF INDIVIDUALS' UTILITY SCORES FOR JOBS AND ID NUMBERS
C       IROW = LIST OF FINAL ASSIGNMENTS
C       KDEM = JOB QUOTAS
C       ICOLMI = MINIMUM NO. OF JOBS WITHIN EACH CATEGORY TO BE ASSIGNED BY OTT
C       KCC = ESTIMATES OF COLUMN CONSTANTS FOR USE IN OTT (0'S IF NOT KNOWN)
C       MEN = NUMBER OF INDIVIDUALS TO BE ASSIGNED , ALSO SUM OF JOB QUOTAS
C       LC = NUMBER OF JOB CATEGORIES TO WHICH THE INDIVIDUALS WILL BE ASSIGNED
C       IDIF = THE LARGEST UTILITY SCORE MUST BE THIS MUCH LARGER THAN THE SECOND
C              LARGEST UTILITY SCORE IN ORDER TO ASSIGN AN INDIVIDUAL TO THE JOB
C              IN PRESEL
C       KDIF = THE MAXIMUM VALUE IDIF CAN TAKE ON.  SET TO IDIF
C               IF ONLY ONE CYCLE IS DESIRED
C       IADDIF = THE AMOUNT BY WHICH IDIF IS INCREASED FOR EACH ALLOCATION CYCLE
C       IOPTIO = OPTIONS FOR ASSIGNMENTS. 1 = ALLOCATION BY PRESEL AND OTT
C               2 = ALLOCATION BY OTT ONLY  3 = RANK ORDER FIRST,THEN
C              ALLOCATE BY PRESEL AND OTT
C       JPRINT = OPTION TO PRINT OUT THE IMPUTTED DATA, THE NON-NORMALIZED
C               UTILITY SCORES, THE RANGES AND MINIMUM FOR CHECKING PURPOSES
C               DURING THE SUBROUTINE INPUT.  1 = PRINT 2 = NO PRINT
C       NAME = MATRIX OF LABELS FOR THE INDIVIDUALS
C       JOBN = NAMES OF THE JOB CATEGORIES
      COMMON KO(200,9)
      DIMENSION JOHN(8)
      COMMON IROW(200)
      COMMON KDEM(8),IDEM(8)
      COMMON ICOLMI (8)
      COMMON KCC(8)
      COMMON MEN,LC,IDIF,KDIF,IADDIF,IOPTIO,JPRINT
      COMMON NAME(200,18)
      DATA (JOBN(IZ),IZ=1,8)/'ROTC','CE','LT','ST','C ','CS','I','RIF'/
C  BRING IN AND SET UP DATA
      CALL INPUT
 4    ID=LC+1
      N=MEN
C  RANK ORDER INDIVIDUALS BY TOTAL UTILITY SCORE, IF WISHED
      IF (IOPTIO.NE. 3) GO TO 8
      CALL RANKR (MEN,ID)
C  GIVE EACH INDIVIDUAL AN ID NUMBER
 8    DO 15  JE=1,MEN
15    KO(JE,ID)=JE
C  DUPLICATE KDEM FOR LATER USE
      DO  21  JE=1,LC
21    IDEM(JE)=KDEM(JE)
      PRINT 77
77    FORMAT(//)
      PRINT  78
78    FORMAT  (27HOQUOTAS TO BE FILLED
      PRINT 775
775   FORMAT (6X,4HROTC,5X,2HCE,5X,2HLT,6X,2HST,6X,2HCC,6X,2HCS,7X1HI,
     C6X,3HRIF)
      PRINT  79,(IDEM(KK),KK=1,LC)
79    FORMAT (8I8)
      PRINT 80,IOPTIO
80    FORMAT(1H0,I3)
      GO TO (86,87,86) ,IOPTIO
86    CONTINUE
C  PRESELECT SOME ASSIGNMENTS
      CALL PRESEL (LC,ID,MEN,IDIF,N)
```

```
C  OPTIMIZE JOB UTILITY SCORES WITH THE REMAINDER
87     CALL OTT  (N,LC)
C  CALCULATE TOTAL UTILITY SCORE
       CALL CALCU (IOPTIO,MEN)
       PRINT 90
90     FORMAT(1H1)
       PRINT 93
93     FORMAT(1X'ASSIGNMENTS BY OPTIMIZATION, OFFICER FOLLOWED BY ASSIGNM
      1ENT'/)
       PRINT 94
94     FORMAT (1X,3HNO.,5X,4HNAME)
       DO 96 JZQ=1,N
       JZHOLD=KO(JZQ,ID)
       IRHOLD=IROW(JZQ)
       PRINT 95,JZHOLD,(NAME(JZHOLD,JZ),JZ=1,3),JOBN(IRHOLD)
95     FORMAT(1X,I3,3X,2A6,A3,2X,A4)
96     CONTINUE
       PRINT 98
98     FORMAT(///1X'ASSIGNMENTS BY PRESELECTION, OFFICER FOLLOWED BY ASSI
      1GNMENT'/)
       N = N+1
       DO 99 JZQ=N,MEN
       JZHOLD=KO(JZQ,ID)
       IRHOLD=IROW(JZQ)
       PRINT 95,JZHOLD,(NAME(JZHOLD,JZ),JZ=1,3),JOBN(IRHOLD)
99     CONTINUE
C  ANOTHER CYCLE OF ALLOCATION CAN BE GONE THROUGH BUT WITH AN INCREASED
C  QUALIFICATION FOR ASSIGNMENT BY PRESEL (IDIF) BY THE AMOUNT (IADDIF) TO
C  A LIMIT (KDIF)
       IDIF=IDIF+IADDIF
       IF(IDIF.GT.KDIF) GO TO 100
       GO TO 8
100    CONTINUE
       STOP
       END
**
```

```
      SUBROUTINE INPUT
C  INPUT READS IN AND SETS UP DATA
C       KDEM = JOB QUOTAS
C       ICOLMI = MINIMUM NO. OF JOBS WITHIN EACH CATEGORY TO BE ASSIGNED BY OTT
C       NSCOR = MATRIX OF NORMALIZED SCORES FOR EACH INDIVIDUAL BY EACH
C                JOB CATERGORY
C       NTABLE = MATRIX OF THE WEIGHTS OF EACH VALUE (TO A POSSIBLE 10) OF
C                THE SIX BACKGROUND VARIABLES FOR EACH OF THE EIGHT JOB
C                CATERGORIES
C       NWEIG = MATRIX OF THE OVERALL IMPORTANCE (OR WEIGHT) OF EACH OF
C                THE 6 BACKGROUND VARIABLES FOR EACH OF THE 8 JOB CATERGORIES
C       NMIN = THE MINIMUM POSSIBLE UTILITY SCORES FOR EACH JOB CATEGORY
C       NORMS = THE MAXIMUM RANGE OF POSSIBLE UTILITY SCORES FOR EACH JOB
C                CATEGORY
C       NAME= MATRIX OF INDIVIDUAL DATA LABELS
C       MANREC = MATRIX OF INDIVIDUAL DATA RECORDS
C       NVAR = THE NUMBER OF POSSIBLE VALUES A BACKGROUND VARIABLE CAN
C                TAKE ON FOR EACH JOB CATEGORY
      COMMON NSCOR(200,9)
      COMMON JROW(200)
      COMMON KDEM(8),ICOL(8)
      COMMON ICOLMI (8)
      COMMON KCC(8)
      COMMON MEN,LC,IDIF,KDIF,IADDIF,IOPTIO,JPRINT
      COMMON NAME(200,18)
      DIMENSION MANREC (200,6)
      DIMENSION NTABLE(6,8,10)
      DIMENSION NORMS(8)
      DIMENSION NWEIG(6,8)
      DIMENSION NMIN(8)
      DIMENSION NVAR(6)
C  IN GOES THE PROGRAM DATA
      READ 300,ICOLMI
  300 FORMAT(8I5)
      READ 1,MEN,LC,IDIF,KDIF,IADDIF,IOPTIO,JPRINT
    1 FORMAT (9I5)
      READ 2, (KCC(I),I=1,LC)
      READ 2, (KDEM(I), I = 1,LC)
      READ 2000, NVAR
 2000 FORMAT (6I5)
      READ 10,(((NTABLE(I,J,K),K=1,10),J=1,8),I=1,6)
   10 FORMAT (10I5)
      READ 4,((NWEIG(I,J),I=1,6),J=1,8)
    4 FORMAT (12I5)
C  FINDS THE RANGE AND MINIMUM VALUE OF THE POSSIBLE UTILITY SCORES
C  FOR EACH JOB CATEGORY
      DO 6000 J=1,8
      MAXTOT=0
      MINTOT=0
      DO 5000 I=1,6
C  THE DEFAULT VALUE IN NTABLE (MATRIX OF WEIGHTS FOR EACH VALUE A
C  BACKGROUND VARIABLE CAN TAKE ON) IS 0.  0 IS ALSO A VALID VALUE.  THE
C  NEXT THREE STATEMENTS MAKE SURE THAT ONLY VALUES AND NOT DEFAULT 0'S
C  ARE ENTERED INTO THE CALCULATION.
      L=NVAR(I)
      ILAMAX=NWEIG(I,J)*NTABLE(I,J,1)
      ILAMIN=ILAMAX
      DO 4000 K=1,L
      IBUF=NWEIG(I,J)*NTABLE(I,J,K)
      IF(IBUF.GT.ILAMAX) GO TO 4300
```

```
      IF(IBUF.LT.ILAMIN) GO TO 4500
      GO TO 4000
 4300 ILAMAX=IBUF
      GO TO 4000
 4500 ILAMIN=IBUF
 4000 CONTINUE
 4001 MAXTOT=MAXTOT+ILAMAX
      MINTOT=MINTOT+ILAMIN
 5000 CONTINUE
      NORMS(J)=MAXTOT-MINTOT+1
      NMIN(J)=MINTOT
 6000 CONTINUE
      KOUNT=31
      LK = 0
      N = MEN
      DO 25 I = 1,N
      DO 25 J = 1,8
   25 NSCOR(I,J) = 0
      DO 100 JK = 1,N
      READ 3,(NAME(JK,JM),JM=1,3),(MANREC(JK,NM),NM=1,6)
    3 FORMAT(2A6,A3,6I1)
C  COMPUTE UTILITY SCORES
C  JOB UTILITY SCORE FOR AN INDIVIDUAL = TAKE EACH VALUE OF THE BACKGROUND
C  VARIABLE TIMES THE OVERALL WEIGHT OF THAT VARIABLE FOR THE JOB AND
C  SUM THEM
      DO 99 J = 1,8
      DO 98 I = 1,6
      LL = MANREC(JK,I)
       IF(LL.NE.0) GO TO 97
      LL=10
   97  CONTINUE
   98 NSCOR(JK,J)=NSCOR(JK,J) + (NTABLE(I,J,LL)*NWEIG(I,J))
   99  CONTINUE
  100 CONTINUE
C  PRINT OUT THE DATA READ IN FOR CHECKING PURPOSES, IF DESIRED
      GO TO (6100,6200),JPRINT
 6100 PRINT 21
      PRINT 3003
 3003 FORMAT(35X,35H    VALUES                          )
      PRINT 44
      PRINT 12,(((NTABLE(I,J,K),K=1,10),J=1,8),I=1,6)
      PRINT 21
      PRINT 3004
 3004 FORMAT(35X,35H      NORMS                         )
      PRINT 44
      PRINT 12,NORMS
      PRINT 44
      PRINT 12,NMIN
      PRINT 13
      PRINT 3005
 3005 FORMAT(35X,35H  WEIGHTS                           )
      PRINT 44
      PRINT 12,((NWEIG(I,J),J=1,8),I=1,6)
      PRINT 13
      PRINT 3000
 3000 FORMAT(12X,30H RAW DATA                     )
      PRINT 44
      PRINT 3001
 3001 FORMAT(4X,46HOFFICER NO.  PREF  CE  COMP  MOP  TIS  ASGS   )
      DO 150 JK=1,N
```

```
150   PRINT 8,JK,(MANREC(JK,NM),NM=1,6)
  8   FORMAT(11X,I3,6I5)
      PRINT 13
      PRINT 3002
 3002 FORMAT(20X,50H   NON STANDARDIZED UTILITY SCORES                 )
      PRINT 44
      PRINT 11,((NSCOR(I,J),J=1,8),I=1,N)
  11  FORMAT(1X,8I8)
C  NORMALIZE UTILITY SCORES FROM 0 TO 1000 USING THE RANGE AND MINIMUM
C  COMPUTED ABOVE
6200  DO 50 J = 1,8
      DO 50 I = 1,MEN
      NSCOR(I,J) = NSCOR(I,J) -NMIN(J)
      NSCOR(I,J) = NSCOR(I,J) * 1000
      NSCOR(I,J) = NSCOR(I,J)/NORMS(J)
  50  CONTINUE
C PRINT OUT NORMALIZED UTILITY SCORES WITH LABELS
      PRINT 21
      PRINT 15
      PRINT 44
      PRINT 14
      PRINT 18
      PRINT 22
      PRINT 23
      DO 96 I =1,N
  6   IF(I.EQ.KOUNT) GO TO 5
      GO TO 96
  5   CONTINUE
      KOUNT =KOUNT+30
      PRINT 21
      PRINT 15
      PRINT 44
      PRINT 14
      PRINT 18
      PRINT 22
      PRINT 23
  96  PRINT 19,I,(NAME(I,J),J=1,3),(NSCOR(I,J),J=1,8)
  19  FORMAT (1X,I3,3X,2A6,A3,2X,8I8)
C  FIND MEANS AND SIGMAS OF UTILITY SCORES FOR EACH JOB CATEGORY
      PRINT 796
 796  FORMAT(1H1,7X,3HJOB,6X,4HMEAN,5X,5HSIGMA)
      DO 799 J = 1,LC
      NSUMX = 0
      NSUMX2 = 0
      DO 797 I = 1,N
      NSUMX = NSUMX+NSCOR(I,J)
 797  NSUMX2=NSUMX2 + NSCOR(I,J)*NSCOR(I,J)
      NAVG = NSUMX/N
      VAR = NSUMX2/N - NAVG*NAVG
      NSIG = SQRT(VAR)
      PRINT 798,J,NAVG,NSIG
 798  FORMAT(1X,3I10)
 799  CONTINUE
      PRINT 44
      PRINT 31
  31  FORMAT(1X,  50HROTC = ROTC INSTRUCTOR, CE = CIVILIAN EDUCATION,  )
      PRINT 32
  32  FORMAT(1X,  50HLT = LONG TOUR, ST = SHORT TOUR, C = COMMAND   ,  )
      PRINT 33
  33  FORMAT(1X,  50HCS = CONUS STAFF,I = INSTRUCTOR-RIF= RED. IN FORCE)
```

```
      FORMAT (8I5)
   12 FORMAT(1X,10I10)
   13 FORMAT (//)
   14 FORMAT(30X,1H1,7X,1H2,7X,1H3,7X,1H4,7X,1H5,7X,1H6,7X,1H7,7X,1H8)
   15 FORMAT(29X,48HOFFICER UTILITY SCORES FOR EIGHT ASSN CATEGORIES)
   18 FORMAT(28X4HROTC,5X2HCE,6X2HLT,6X2HST,6X2HC  ,6X2HCS,7X1HI,6X3HRIF)
   21 FORMAT (1H1,/)
   22 FORMAT(1X,3HMAN)
   23 FORMAT(1X,3HNO.,5X,4HNAME)
   44 FORMAT(/)
      RETURN
      END
••
```

```
      SUBROUTINE RANKR (MEN,ID)
C  RANKR IS AN OPTIONAL SUBROUTINE CALLED ONLY IF IOPTIO = 3 IN THE SECOND
C  DATA CARD
C  RANKR RANK ORDERS THE INDIVIDUAL DATA ACCORDING TO TOTAL UTILITY SCORES
C  ACROSS ALL CATEGORIES
C       NSCOR = MATRIX OF NORMALIZED SCORES FOR EACH INDIVIDUAL BY EACH
C               JOB CATERGORY
C       NSTOR = STORAGE MATRIX USED FOR PUTTING NSCOR IN RANK ORDER
      COMMON NSCOR(200,9)
      COMMON IROW(200)
      COMMON KDEM(8),IDEM(8)
      COMMON ICULMI (8)
      COMMON KCC(8)
      COMMON MON,LC,IDIF,KDIF,IADDIF,IOPTIO,JPRINT
      COMMON NAME(200,18)
      DIMENSION NN(200),NNUM(200)
      DIMENSION NSTOR(200,9)
      DIMENSION NAMEST(200,18)
      N=MEN
C  GIVE EACH PERSON AN ID NUMBER
      DO 20 I = 1,N
 20   NN (I) = I
C  DETERMINE A SINGLE UTILITY SCORE FOR EACH PERSON BY ADDING THE FIRST
C  SEVEN JOB UTILITY SCORES.  THE EIGHTH SCORE (FOR RIF) IS A LOGICALLY
C  NEGATIVE SCORE AND IS REFLECTED IN THE OTHER SCORES.  HENCE IT IS NOT
C  INCLUDED IN THIS SUMMATION.
      DO 100 I = 1,N
      DO 100 J = 1,7
 100  NNUM(I)=NNUM(I) +NSCOR(I,J)
C  PRINT MAN NUMBER FOLLOWED BY UTILITY SCORE COMPUTED ABOVE
      PRINT 33
      DO 79 I = 1,N
      PRINT  59, NN(I) , NNUM(I)
 59   FORMAT(1X,I3,5X,I9)
 79   CONTINUE
C  RANK ORDER THE UTILITY SCORES
      DO 65 I = 1,N
      K=I
      DO 60 J =I,N
      IF(NNUM(K).LT.NNUM(J)) K=J
C  LARGER UTILITY SCORE
 60   CONTINUE
      IF(K.EQ.I) GO TO 65
C  SWITCH
      NNN = NNUM(K)
      NNUM(K) = NNUM(I)
      NNUM (I) = NNN
C  ID OF PERSON HAVING LARGERST UTILITY SCORE
C  SWITCH
      NR = NN(K)
      NN(K) = NN(I)
      NN(I) = NR
 65   CONTINUE
C  PRINT LARGEST TO SMALLEST
      PRINT 33
      DO 89 I = 1,N
      PRINT  59, NN(I) , NNUM(I)
 89   CONTINUE
C  NOW RANK ORDER THE UTILITY SCORE MATRIX AND PRINT IT OUT
C  ALSO RANK ORDER THA NAME MATRIX
```

```
      II= ID
      DO 50 I = 1,N
      DO 50 J = 1,II
      MM = NN(I)
 50   NSTOR(I,J) = NSCOR(MM,J)
      DO 51 I = 1,N
      DO 51 J = 1,15
      MM = NN(I)
 51   NAMEST(I,J) = NAME(MM,J)
      DO 70 I = 1,N
      DO 70 J = 1,II
 70   NSCOR(I,J) = NSTOR(I,J)
      DO 71 I = 1,N
      DO 71 J =1,15
 71   NAME(I,J) = NAMEST(I,J)
      PRINT 402
 402  FORMAT(16H1 RANKED MATRIX      )
      PRINT 401
 401 0FORMAT(1X,100H NEW NO.
     1                                       OLD NO.
      DO 91 I = 1,N
 91   NSCOR(I,II) = NN(I)
      DO 92 I = 1,N
      PRINT 403, I, (NSCOR(I,J),J=1,II)
 92   CONTINUE
      DO 7707 I = 1,N
      PRINT 404,I, NSCOR(I,ID)
 404  FORMAT(1H ,2I5)
 7707 CONTINUE
 403  FORMAT(1X,I3,9I9)
 33   FORMAT(///)
      RETURN
      END
```

```
      SUBROUTINE PRESEL(K,IO,MEN,IDIF,N)
C  PRESEL PRESELCTS THE ASSIGNMENTS OF A CERTAIN NUMBER OF PEOPLE BEFORE
C  ASSIGNING THE REST BY OTT.  IT ASSIGNS PEOPLE TO A JOB CATEGORY (UP TO A
C  PREDETERMINED LIMIT) IF THEIR HIGHEST SCORE IS GREATER THAN THEIR SECOND
C  HIGHEST SCORE BY A CERTAIN AMOUNT (IDIF).  HOWEVER IF THERE AREN'T
C  SUFFICIENT PEOPLE WITH APPROPRIATE SCORES, PRESEL DOES NOT ATTEMPT TO
C  MEET THE LIMITS ( IDEM = ICOLMI).
C      K = LC, NO. OF JOB CATEGORIES
C      N = MEN, STARTING VALUE
C      N = NUMBER OF PEOPLE LEFT TO BE ASSIGNED BY OTT, FINAL VALUE
C      KO = NSCOR, MATIX OF UTILITY SCORES
C      KDEM = JOB QUOTAS
C      IDEM = KDEM, STARTING VALUE
C      IDEM = JOB QUOTAS TO BE FILLED BY OTT, FINAL VALUE
C      IROW = LIST OF FINAL ASSIGNMENTS
C      ICOLMI = MINIMUM NO. OF JOBS WITHIN EACH CATEGORY TO BE ASSIGNED BY OTT
C      IDIV = NUMBER OF PEOPLE ASSIGNED BY PRESEL
      COMMON KO(200,9)
      COMMON IROW(200)
      COMMON  KDEM(8),IDEM(8)
      COMMON ICOLMI (8)
      PRINT  96, IDIF
96    FORMAT (40HOPRIOR TO OTT,ASSIGNING DIFFERENCES OF    I4)
      PRINT  62, ICOLMI
62    FORMAT  (46HOQUOTA RESTRICTION FOR ASSIGNMENT PRIOR TO OTT/8I6)
      PRINT 20
20    FORMAT(1H1)
      PRINT 24
24    FORMAT(1X,'OFFICERS ASSIGNED BY PRESELECTION',32X,'OFFICERS TO BE
     1ASSIGNED BY OPTIMIZATION'/6X,'HIGHEST SCORES PRINTED',43X,'HIGHEST
     2 SCORES PRINTED')
      PRINT 27
27    FORMAT(1HO,'NO.    HIGHEST    2ND HIGHEST',34X,'NO.    HIGHEST
     1    2ND HIGHEST')
C  FIND THE TWO LARGEST UTILITY SCORES FOR AN INDIVIDUAL
C      IO = LARGEST SCORE
C      IT = SECOND LARGEST SCORE
      J = 0
      DO 95 KKK=1,MEN
      J = J+1
      IF  (KO(J,1).GE.KO(J,2))GO TO 81
      GO TO 93
81    IO=1
      IT=2
      GO TO 92
93    IO=2
      IT=1
92    DO  88  L=3,K
      IF   (KO(J,IT).GE.KO(J,L))GO TO 88
91    IF   (KO(J,IO).GT.KO(J,L))GO TO 90
      GO TO 89
90    IT=L
      GO TO 88
89    IT=IO
      IO=L
88    CONTINUE
C  IF THE LARGEST SCORE IS GREATER THAN THE SECOND LARGEST BY A PREDETERMINED
C  AMOUNT (IDIF) AND THERE IS STILL A JOB AVAILABLE IN THE APPROPRIATE
C  JOB CATEGORY...
      IF(KO(J,IO).GE.(KO(J,IT)+IDIF).AND.IDEM(IO).G"`ICOLMI(IO))GOTO 87
```

```
      GO TO 94
87    CONTINUE
C  THEN MAKE THE ASSIGNMENT BY PRINTING OUT THE TWO HIGHEST SCORED JOB
C  CATEGORIES AND ...
      PRINT 75,KO(J,9),IO,IT
75    FORMAT( 1X,I3,6X,I2,12X,I2)
C  ...SWITCHING THE INDIVIDUAL'S SCORES WITH
C  AN UNTESTED INDIVIDUAL'S, THUS BUILDING  UP A LIST OF ASSIGNED PEOPLE
C  FROM THE BOTTOM OF KO AND LEAVING THE PEOPLE TO BE ASSIGNED BY OTT AT
C  THE TOP.
      DO 4 J1=1,ID
      IHOLD=KO(J,J1)
      KO(J,J1)=KO(N,J1)
4     KO(N,J1)=IHOLD
C  DECREMENT THE APPROPRIATE JOB QUOTA
      IDEM(IO)=IDEM(IO)-1
C  ENTER THE ASSIGNMENT IN IROW
      IROW(N)=IO
C  DECREMENT THE INDEX AND THE NUMBER OF PEOPLE TO BE CONSIDERED
      J=J-1
      N=N-1
      GO TO 95
C  OTHERWISE JUST PRINT OUT THE SCORES
94    PRINT 99,KO(J,9),IO,IT
99    FORMAT(64X,I3,6X,I2,12X,I2)
95    CONTINUE
100   PRINT 19
      PRINT 19
      IDIV=MEN-N
      PRINT  84,  IDIV
84    FORMAT (34H0OFFICERS ALLOCATED PRIOR TO OTT =I4)
      PRINT  82,N
82    FORMAT  (33H0OFFICERS TO BE ALLOCATED BY OTT=I4)
      PRINT  79
79    FORMAT  (36H0RESIDUAL QUOTAS TO BE FILLED BY OTT)
      PRINT  78,(IDEM(KK),KK=1,K)
78    FORMAT  (8I8)
19    FORMAT(/)
      RETURN
      END
**
```

```
      SUBROUTINE OTT (N,K)
C IC  MUST  BE  DIMENSIONED  EQUAL  TO  THE  SCORE  MATRIX  IN  THE  CALLING
C PROGRAM
C IROWC  MUST  BE  DIMENSIONED  AT  LEAST  AS  LARGE  N
C ICOLC  MUST  BE  DIMENSIONED  AT  LEAST  AS  LARGE  AS  K
C          IC = MATRIX  OF  APTITUDE  AREA  SCORES
C          N = NUMBER  OF  MEN  TO  BE  ASSIGNED
C          K = NUMBER  OF  JOB  CLASSIFICATIONS
C          ICOL = QUOTAS  FOR  EACH  JOB    SUM  OF  QUOTAS  MUST  EQUAL  N
C          JCOLC = ESTIMATE  OF  COLUMN  CONSTANTS
C
C          IF  ACCURATE  ESTIMATES  CAN  BE  MADE  FEWER  ITERATIONS  ARE
C          REQUIRED  AND  A  SUBSTANTIAL  IMPROVEMENT  IN  EFFICIENCY (SPEED
C          IN  REACHING  A  SOLUTION) RESULTS
C
C          IF  NO  ESTIMATE  IS  AVAILABLE  JCOLC  SHOULD  BE  SET  EQUAL
C          TO  ZERO  BEFORE  CALLING  OTT
C          JROW = SET  OF  ASSIGNMENTS  CORRESPONDING  TO  OPTIMAL  ALLOCATION
C      RETURNED  TO  MAIN  PROGRAM
C
C THE  SOLUTION  CAN  ALWAYS  BE  CHECKED  BY  ADDING  THE  COLUMN  CONSTANTS
C TO  THE  ORIGINAL  MATRIX  OF  PERFORMANCE  ESTIMATES//THE  ADJUSTED  SCORE
C FOR  THAT  CATEGORY  TO  WHICH  EACH  INDIVIDUAL  IS  ASSIGNED  WILL  BE  AS
C HIGH  OR  HIGHER  THAN  ALL  OTHER  ADJUSTED  PERFORMANCE  ESTIMATES  FOR
C THAT  MAN
      COMMON IC(200,9)
      COMMON JROW(200)
      COMMON KDEM(8),ICOL(8)
      COMMON ICOLMI (8)
      COMMON JCOLC(8)
      COMMON MEN,LC,IDIF,KDIF,IADDIF,IOPTIO,JPRINT
      DIMENSION IROWC(200),ICOLC(8)
C
C          ADJUSTING  APTITUDE  SCORES  BY  ESTIMATED  COLUMN  CONSTANTS  AND
C          SETTING  ALL  SCORES  AS  A  POSITIVE  DIFFERENCE  FROM  THE
C          MAXIMUM  SCORE
      DO 105 I=1,N
      IC(I,1)=IC(I,1)+JCOLC(1)
      MAX=IC(I,1)
      DO 104 J=2,K
      IC(I,J)=IC(I,J)+JCOLC(J)
      IF(MAX.LT.IC(I,J))GO TO 103
      GO TO 104
  103 MAX=IC(I,J)
  104 CONTINUE
      IROWC(I)=MAX
      DO 105 J=1,K
  105 IC(I,J)=MAX-IC(I,J)
C          CHECK  TO  DETERMINE  IF  SUM  OF  QUOTAS  EQUALS  NUMBER  OF
C          MEN  TO  BE  ASSIGNED
      ISMASG=0
      DO 1 J = 1,K
      ISMASG=ISMASG + ICOL(J)
    1 ICOLC(J) = 0
      IF(ISMASG.EQ.N)GO TO 1001
      GO TO 50
 1001 DO 2 I=1,N
    2 JROW(I) = 1
    3 MM = 0
    4 DO 31 I=1,N
```

```
      IF(JROW(I))31,5,5
    5 DO 30 J=1,K
      IF(ICOL(J))30,6,6
    6 IF(IC(I,J))7,8,30
    7 JROW(I) = -1
      MM=1
      GO TO 31
    8 MM=1
      IF(JROW(I))52,9,13
    9 JROW(I) = -1
      IC(I,J) = -1
      DO 12 JJ=1,K
      IF(IC(I,JJ) + 2)52,10,12
   10 IF(ICOL(JJ))11,12,12
   11 ICOL(JJ) = 0
      JROW(I) = -2
   12 CONTINUE
      GO TO 31
   13 JROW(I) = 0
      IC(I,J) = -2
      JJ = J
      M = I
   14 IF(ICOL(JJ))52,15,22
   15 DO 18 II=1,N
      IF(M-II)16,18,16
   16 IF(IC(II,JJ) + 2)52,17,18
   17 IF(JROW(II) + 2)52,19,18
   18 CONTINUE
   19 IC(II,JJ) = 0
      DO 20 JJ = 1,K
      IF(IC(II,JJ) + 1)52,21,20
   20 CONTINUE
   21 IC(II,JJ) = -2
      JROW(II) = -1
      M = II
      GO TO 14
   22 ICOL(JJ) = ICOL(JJ) - 1
      DO 26 II = 1,N
      IF(JROW(II))23,24,24
   23 JROW(II) = 0
   24 DO 26 JJ=1,K
      IF(IC(II,JJ) + 1)26,25,26
   25 IC(II,JJ) = 0
   26 CONTINUE
      IND = 0
      DO 29 JJ = 1,K
      IF(ICOL(JJ))28,27,29
   27 ICOL(JJ) = -1
   28 IND = IND + 1
   29 CONTINUE
      IF(K-IND)52,54,31
   30 CONTINUE
   31 CONTINUE
      IF(MM)52,32,3
   32 IMIN = 9999
      DO 38 I=1,N
      IF(JROW(I))38,33,33
   33 DO 37 J=1,K
      IF(ICOL(J))37,34,34
   34 IF(IMIN-IC(I,J))37,37,35
```

```
   35 IF(IC(I,J))37,52,36
   36 IMIN = IC(I,J)
   37 CONTINUE
   38 CONTINUE
C TRANSFORMING  THE  MATRIX  BY  ADDING  CONSTANTS  TO  APPROPRIATE  COLUMNS
      DO 42 I=1,N
      IH = 0
      IF(JROW(I))39,40,40
   39 IH = IMIN
   40 DO 42 J =1,K
      IG = 0
      IF(ICOL(J))42,41,41
   41 IG = -IMIN
   42 IC(I,J) = IC(I,J) + IH + IG
      NUMITE =NUMITE +1
      DO 44 J = 1,K
      IF(ICOL(J))44,43,43
   43 ICOLC(J) = ICOLC(J) + IMIN
   44 CONTINUE
      GO TO 4
   50 PRINT 51,ISMASG,N,ICOL
   51 FORMAT (49H SIZE  OF  POOL  INCONSISTENT  WITH  ASSIGNMENTS //2I12
     1//8I10)
      GO TO 60
   52 PRINT 53
   53 FORMAT (6H ERROR)
   54 DO 58 II=1,N
      MIMI=9999
      DO 56 JJ=1,K
      IF(IC(II,JJ) + 2)58,55,57
   55 IC(II,JJ) = 0
      JROW(II) = JJ
57    IROAD=IC(II,JJ)+ICOLC(JJ)+JCOLC(JJ)
      IC(II,JJ)=IROWC(II)-IROAD
      IF (IROAD.GE.MIMI) GO TO 56
5656  MIMI=IROAD
56    CONTINUE
      DO  5657  L=1,K
5657  IC(II,L)=IC(II,L)+MIMI
   58 CONTINUE
      DO 581 JJ=1,K
  581 ICOLC(JJ)=ICOLC(JJ)+JCOLC(JJ)
C PRINTING  SET  OF  ALLOCATION  CRITERIA
      PRINT 59,(ICOLC(JJ),JJ=1,K)
   59 FORMAT (1X30 H SET  OF  ALLOCATION  CRITERIA ,10I9)
   60 RETURN
      END
**
```

```
      SUBROUTINE CALCU(IOPTIO,MEN)
C  CALCU CALCULATES THE ALLOCATION AVERAGE FOR A SET OF ASSIGNMENTS
      COMMON KO(200,9)
      COMMON IROW(200)
      BSUM=0.0
      DO 9   JOR=1,MEN
      IOR=IROW(JOR)
9     BSUM=KO(JOR,IOR)+BSUM
      AVE=BSUM/MEN
      GO TO (42,62,42),IOPTIO
42    PRINT  43,AVE
43    FORMAT  (50H0 ALLOCATION AVERAGE FOR MODOTT, FOR SAMPLE INPUT=F8.3)
     1)
      GO TO 10
62    PRINT  63,AVE
63    FORMAT  (46H0ALLOCATION AVERAGE FOR OTT, FOR SAMPLE INPUT=F8.3)
      RETURN
10    END
**
```